



Arduino Theremin

Written By: Alan

TOOLS:

- [Hot Glue gun & hot glue \(1\)](#)
- [Soldering iron \(1\)](#)
- [Super glue \(1\)](#)

PARTS:

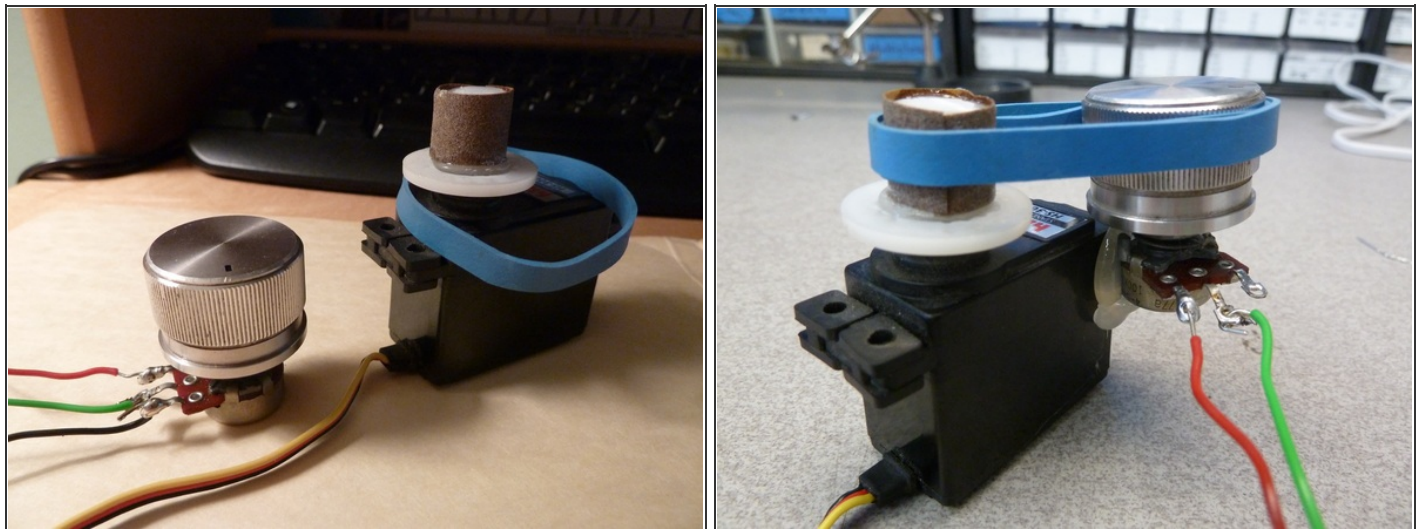
- [Arduino microcontroller, Uno or Duemilanove \(1\)](#)
- [Breadboard kit \(1\)](#)
- [3-16V Piezo Buzzer \(RadioShack #273-074\) \(1\)](#)
- [VEX Ultrasonic Range Finder \(1\)](#)
- [CdS Photoresistor \(1\)](#)
- [10K Resistor \(1\)](#)
- [Potentiometer, 5k \$\Omega\$, audio taper \(1\)](#)
- [Servo \(generic\) \(1\)](#)
- [Circuit board header: 3-pin \(1\)](#)
- [Breadboard jumper wires, or solid core 22AWG wire \(Many\)](#)
Many colours
- [Wire \(A few\)](#)
Many Colours
- [Rubber bands \(A few\)](#)
- [Dowel, 3/8" dia. 1/2" length \(1\)](#)
Or near
- [Sandpaper \(1\)](#)
Strip, 1/2" x 2"

SUMMARY

If you want to make music, then this project is for you. This is a theremin-like device to

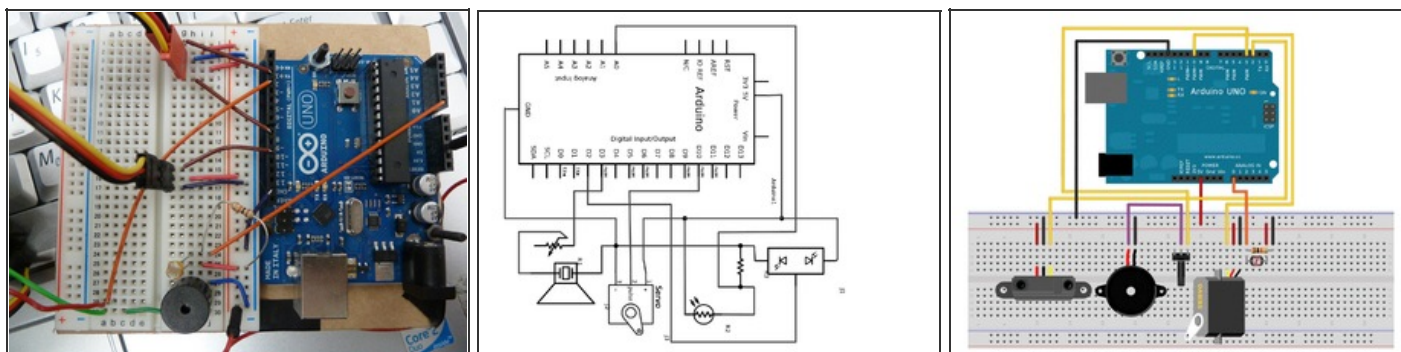
control the pitch and volume of a musical note. It utilizes a photo-resistor to control the pitch, and an ultrasonic sensor to control a servo which turns a potentiometer controlling the volume.

Step 1 — Connect the servo and potentiometer.



- Solder wires onto the potentiometer. I used red for the audio voltage in on the left and green for the voltage out in the center.
- I initially soldered on a black wire on the third pin, but removed it as it was redundant.
- Hot glue the dowel onto the servo, centered, and then super glue the sandpaper to the dowel to provide traction for the rubber band.
- Find a rubber band, preferably wide, that can be used to transfer the servo motion to the potentiometer.
- Hot glue the potentiometer in place. Don't be sparing with the hot glue, as it is not terribly strong.
- Turn both the potentiometer and servo all the way to the left and link them together with the rubber band. I had to use several as the servo shifted while the glue was cooling.

Step 2 — Wire that Arduino.



- Supply +5V to the power rail on the breadboard, and then to the power pin on each ultrasonic sensor, the servo power pin, and the photo-resistor.
- Supply GND to the GND rail on the breadboard, and then to the ground pin on each ultrasonic sensor, the servo ground pin, and the piezo buzzer ground pin.
- I used analog 0 for the photo-resistor output, digital 10 for the servo, digital 2 for the ultrasonic output, and digital 8 for the ultrasonic input
- Use a 10kΩ resistor as a pull-down for the photo-resistor by connecting it to GND and analog 0, or whatever analog pin you end up connecting the photo-resistor to.

Step 3 — Code the Arduino.

```

theremin | Arduino 1.0.1
File Edit Sketch Tools Help
theremin.ino

/*
  theremin.ino

  This is a theremin-like device, powered by an arduino. It utilizes
  an ultrasonic sensor to control the volume, and a photoresistor to
  control the pitch. The pitch is controlled directly by the
  Arduino, whereas the volume is controlled by a servo changing the
  position of a potentiometer, increasing or decreasing the voltage
  sent to the speaker.

  Circuit Schematic:
  https://drive.google.com/file/d/0Bz3kxtp-0j-ua21000Y0hV69Xk
  Ultrasonic Input -- D8
  Ultrasonic Output -- D2
  Speaker + -- D3
  Servo pulse -- B10
  Photoresistor -- A0

  Alan Grier
  12/3/2012
  */

#include <Servo.h> //standard arduino servo library
#include <Ultrasonic.h> //standard arduino ultrasonic library

Ultrasonic ultrasonic(8,2); //sets input pin as 8, and output pin as 2
Servo servo; //assigns variable servo to my servo

void setup()
{
  Serial.begin(9600); //initializes serial monitor for troubleshooting
  Serial.println("testing...");

  servo.attach(10); //assigns pin for servo
}

void loop()
{
  int dist = 25; //assign variable and dummy value for ultrasonic value
  dist = ultrasonic.Ranging(CM); //fetches distance from ultrasonic in centimeters

  int light = 512; //assign variable and dummy value for photoresistor value
  int photores = 0; //assign photoresistor pin
  
```

- See attached file, `theremin.pdf`, for the code.
- The volume should increase as your hand is moved closer to the ultrasonic sensor, so if the opposite happens, simply reverse the mapping of the servo position.
- To do this, change the code from `rad = map(dist, 8, 20, 0, 90);` to `rad = map(dist, 8, 20, 90, 0);`